Paper 290-2010

Improving Your Statistical Consulting Prowess by Adding R to Your SAS® Repertoire

Jeff Gossett, University Arkansas Medical Sciences, Department of Pediatrics, Little Rock, AR
Mark Austen, Veterans Administration Health Sciences Research and Development, North Little Rock, AR
Maria Melguizo, University Arkansas Medical Sciences, Department of Pediatrics, Little Rock, AR
Chan-hee Jo, University Arkansas Medical Sciences, Department of Pediatrics, Little Rock, AR

ABSTRACT

SAS® is a powerful package with tremendous capabilities. However, no package contains every model, calculation, or graphics option. With SAS® 9.2, it is easier to tap into the R programming environment with the SAS/IML® Workshop 3.2 application. Many cutting-edge models and statistical graphics are being developed in R. We will discuss strategies for using R to complement a SAS analysis. We will discuss examples where we have used R to solve problems that we could not have handled easily with SAS alone.

INTRODUCTION

Statistical software has experienced explosive growth in features and capabilities recently. Due to the diversity of the new methods being developed, it is exceedingly difficult for a single statistical software package to provide every solution needed by the statistical consultant. Recently R has become the program of choice for many academic researchers to develop new statistical tools. SAS has made it easier to interact with the R package with the new SAS/IML Workshop 3.2. Because R is an open source program, and amateurs can develop new packages independently, new packages can be developed rapidly. However, there is an understandable lack of consistency in the validation, testing, documentation, and formatting of results. We discuss strategies for using R to complement a SAS analysis. We will discuss examples where we have used R to solve problems that we could not have handled easily with SAS alone.

STRATEGIES FOR USING R WITH SAS

There are many strategies for incorporating R into a SAS analysis. Perhaps the simplest is to run each program independently. You can certainly export a SAS dataset to a comma delimited file, and import the resulting file into R. Text output from R can be saved using the sink command. Graphs can be saved as jpeg files. Additional strategies are described in a paper by Philip Holland entitled "SAS to R to SAS."

In SAS 9.2, there is a new option. SAS IML Workshop 3.2 makes it easier to outsource calculations to R. With SAS/IML Studio 3.2, you can easily export SAS permanent data sets to R dataframes with the `ExportDataSetToR` command. If R program output can be coerced into either a matrix or dataframe, it is easy to convert back to a SAS IML matrix of data set. R code is submitted in batches that start with "submit /R" and end with "endsubmit. Text output and graphs from the R code appear in the IML Workshop output windows by default.

Features are added to base R by installing packages. As of February 28, 2010, the CRAN package repository featured 2223 packages (http://cran.r-project.org/web/packages/). Some of the packages that we use regularly include: `foreign` (data transfer), `Design` (over 200 functions for model fitting, testing, estimation, validation, graphics, prediction, and typesetting.), `Hmisc` (functions for data reduction, imputation, power and sample size calculation, advanced table making, recoding variables, recoding variables, importing data, and general graphics), `cmprsk` (fits Fine Gray models for competing risks and implements Gray's tests), `logistf` (fits Firth's penalized log likelihood logistic regression models to combat quasi and complete separation problems in logistic regression as an alternative to exact logistic regression), `ggplot` (graphics), and `lattice` (graphics). Note that SAS added a Firth logistic regression option to PROC LOGISTIC in version 9.2. SAS has also made recent improvements to graphics with the SGPLOT and SGPANEL procedures.

Note that before you can use R within the SAS IML Workshop, you have to install R on the same computer and install any packages that you might want to use. It may be helpful to test the R code within an R session first and ensure that necessary packages are installed. To run R commands in IML Workshop: R commands are submitted in batches starting with the line: "`submit /R;`" and ending with "`endsubmit;.`" Lines must end with semicolons. Comments start with # rather than *. Some R functions seem to have a problem when variable names containing

underscores. Typically you create a SAS data set within your SAS session. You use a libname statement in IML Workshop to access data sets created by SAS. Use the command:

```
run ExportDataSetToR("libname.dsname", "r.data.frame.name");
```

to convert a SAS data set to an R data frame. As a simple example, the following program transfers a data set from the SAShelp library into an R data frame named df. The `names(df)` command asks R to display the names of the variables in the data frame. Code submitted in the SAS/IML Workshop 3.2.

```
run ExportDataSetToR("Sashelp.Class", "df");
submit /R;
    names(df);
endsubmit;
```

EXAMPLES OF USING R

A series of examples will be used to demonstrate calculations in R.

EXAMPLE 1 – THE UBIQUITOUS SUMMARY TABLE

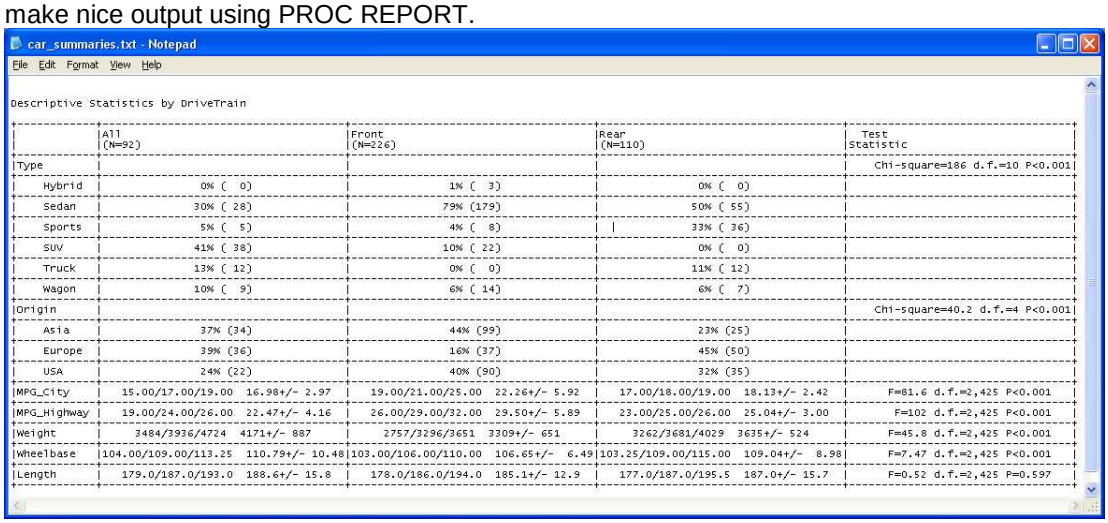
Most manuscripts have a table to summarize risk factors and/or demographics variables by treatment group. Continuous variables are summarized by either mean (SD) or quartiles (q1, q2, q3). Categorical variables are summarized by percentage (n/N). For each factor or variable there is a p-value to assess treatment differences. In order to build this table in SAS, output could be combined from some combination of the TABULATE, MEANS, T-TEST, NPAR1WAY, and FREQ procedures. In R, one could use the `summary.formula()` function from the Hmisc package to make the table. By default, continuous variables are summarized by the triplet of quartiles, q1, q2, q3. The `prmsd=TRUE` option requests the mean+/-SD, too. Options can be used to customize the format of the table. A good introduction to this function can be found in the *Introduction to R notes* by Teresa Scott. For this example we will use the Cars dataset that can be found in the SASHELP data library. Note that variable names in R are case sensitive. The "names" function in R can be used to reveal the variable names of a data frame "`names(cars);`".

We will summarize the variables: Type, Origin, MPG_City, MPG_Highway, Weight, Wheelbase, and Length by DriveTrain. The variables are entered in the form of an equation: group ~ x1 + x2 + … The "`overall=FALSE`" option controls whether you want overall summaries. The "`test=TRUE`" option controls whether you get statistical comparisons and p-values. The "`prmsd=TRUE`" controls whether you get the mean and standard deviation as part of the summary for continuous variables. By default, the continuous variables are tested using the Wilcoxon rank-sum tests (2 groups) or Kruskal-Wallis tests (>2 groups). Categorical variables are compared using Pearson's chi-square test of association.

Code submitted in the SAS/IML Workshop 3.2.

```
# Export this dataset to R using the command:
run ExportDataSetToR("Sashelp.cars", "cars");
submit /R;
library(Hmisc, T);
# send output to external text file. # note use of forward slashes;
sink("c/sugi2010/car_summaries.txt");
#create object named car.sum containing summary object.;
  car.sum <- summary.formula(DriveTrain ~ Type + Origin + MPG_City + MPG_Highway +
Weight + Wheelbase + Length, data = cars, method = "reverse", overall = FALSE, test
= TRUE,digits = 3, prmsd=TRUE, exclude1 = FALSE, long = TRUE);
# use print command with the sink command to send output to an external text file. ;
  print(car.sum, test = TRUE,digits = 3, prmsd=TRUE, exclude1 = FALSE, long = TRUE);
sink();
endsubmit;
```

Figure 1: Screen shot of Descriptive Statistics by DriveTrain. The format is not nice, and the output object *car.sum* cannot be coerced into into a data frame or matrix. However, the results are in a format that are easy to read into SAS using a DATA step. Once we have the results as a SAS data set, we can make nice output using PROC REPORT.



car_summaries.txt - Notepad

File Edit Format View Help

Descriptive Statistics by DriveTrain

	All (N=92)	Front (N=226)	Rear (N=110)	Test Statistic
Type				Chi-square=186 d.f.=10 P<0.001
Hybrid	0% (0)	1% (3)	0% (0)	
Sedan	30% (28)	79% (179)	50% (55)	
Sports	5% (5)	4% (8)	33% (36)	
SUV	41% (38)	10% (22)	0% (0)	
Truck	13% (12)	0% (0)	11% (12)	
Wagon	10% (9)	6% (14)	6% (7)	
Origin				Chi-square=40.2 d.f.=4 P<0.001
Asia	37% (34)	44% (99)	23% (25)	
Europe	39% (36)	16% (37)	45% (50)	
USA	24% (22)	40% (90)	32% (35)	
MPG_City	15.00/17.00/19.00 16.98+/- 2.97	19.00/21.00/25.00 22.26+/- 5.92	17.00/18.00/19.00 18.13+/- 2.42	F=81.6 d.f.=2,425 P<0.001
MPG_Highway	19.00/24.00/26.00 22.47+/- 4.16	26.00/29.00/32.00 29.50+/- 5.89	23.00/25.00/26.00 25.04+/- 3.00	F=102 d.f.=2,425 P<0.001
Weight	3484/3936/4724 4171+/- 887	2757/3296/3651 3309+/- 651	3262/3681/4029 3635+/- 524	F=45.8 d.f.=2,425 P<0.001
Wheelbase	104.00/109.00/113.25 110.79+/- 10.48	103.00/106.00/110.00 106.65+/- 6.49	103.25/109.00/115.00 109.04+/- 8.98	F=7.47 d.f.=2,425 P<0.001
Length	179.0/187.0/193.0 188.6+/- 15.8	178.0/186.0/194.0 185.1+/- 12.9	177.0/187.0/195.5 187.0+/- 15.7	F=0.52 d.f.=2,425 P=0.597

SAS code to read in text file with summaries, and print using PROC REPORT:

```
data cars2;
  infile "("c/sugi2010/car_summaries.txt"" dsd dlm="|"
         lrecl=1024 firstobs=5 truncover;
  format junk $20. label $34. all $34. front $35. rear $35. test $35. ;
  input junk $ label $ all $ front $ rear $ test $ ;
  if substr(junk,1,1)="+" then delete;
  drop junk;
  run;

ods rtf file="c:\sugi2010\sas_r\car_summaries2_09oct20b.rtf" ;
title "Summaries by treatment group";
proc report data=cars2 nowindows;
  columns label all front rear test;
  run; ods rtf close;
```

Although the column headings need to be cleaned up (first 3 rows), the output is much nicer in PROC REPORT.

Table 1: PROC REPORT output. A clever programmer could combine the first 3 rows of the table into a single row. Or the names could be edited manually in a word processor.

label	all	front	rear	test
	All	Front	Rear	Test
	(N=92)	(N=226)	(N=110)	Statistic
Type				Chi-square=186 d.f.=10 P<0.001
Hybrid	0% (0)	1% (3)	0% (0)	
Sedan	30% (28)	79% (179)	50% (55)	
Sports	5% (5)	4% (8)	33% (36)	
SUV	41% (38)	10% (22)	0% (0)	
Truck	13% (12)	0% (0)	11% (12)	
Wagon	10% (9)	6% (14)	6% (7)	

label	all	front	rear	test
Origin				Chi-square=40 d.f.=4 P<0.001
Asia	37% (34)	44% (99)	23% (25)	
Europe	39% (36)	16% (37)	45% (50)	
USA	24% (22)	40% (90)	32% (35)	
MPG_City	15.0/17.0/19.0 17.0+/- 3.0	19.0/21.0/25.0 22.3+/- 5.9	17.0/18.0/19.0 18.1+/- 2.4	F=82 d.f.=(2,425) P<0.001
MPG_Highway	19.0/24.0/26.0 22.5+/- 4.2	26.0/29.0/32.0 29.5+/- 5.9	23.0/25.0/26.0 25.0+/- 3.0	F=102 d.f.=(2,425) P<0.001
Weight	3484/3936/4724 4171+/- 887	2757/3296/3651 3309+/- 651	3262/3681/4029 3635+/- 524	F=46 d.f.=(2,425) P<0.001
Wheelbase	104.0/109.0/113.2 110.8+/- 10.5	103.0/106.0/110.0 106.7+/- 6.5	103.2/109.0/115.0 109.0+/- 9.0	F=7.5 d.f.=(2,425) P<0.001
Length	179/187/193 189+/- 16	178/186/194 185+/- 13	177/187/196 187+/- 16	F=0.52 d.f.=(2,425) P=0.6

Continuous variables are summarized by the triplet q1, q2, q3, followed by the mean+/-SD. Categorical variables are summarized with %(n). A Pearson chi-square test is used to test for association of categorical variables. A Wilcoxon test is used to compare continuous variables. Unfortunately, if your client does not want to see the quartiles, they will have to be removed in a SAS data step.

EXAMPLE 2: PLOTTING MEANS WITH USER CALCULATED ERROR BARS. IN THIS EXAMPLE, SOME SIMPLE DATA CALCULATIONS ARE DEMONSTRATED.

Start with the Cars data set. Convert SAS data set to dataframe named “cars”. Make Hmisc library/package available with library statement. Create dataframe named cars2 that is a subset of cars. The “is.na” function is useful for detecting missing values. The “!” is the “NOT” operator. The cars2 dataframe drops records that are missing the number of cylinders and cars with 3 and 5 cylinders. Create dataframes named mean.hw and mean.city containing summaries and bootstrap generated 95% confidence intervals for mean. Before combining data frames with the rbind function, both data frames need to have common variable names. Finally, plot means and confidence intervals using the xYplot command.

R Code submitted in the SAS/IML Workshop 3.2.

```
run ExportDataSetToR("Sashelp.CARS", "cars");
 submit /R;
   library(Hmisc, T);
## Create data frame – cars2 – dropping cars with 3, missing, and 5 cylinders;
cars2 <- cars[cars$Cylinders>3 & !is.na(cars$Cylinders) & cars$Cylinders != 5,]
## place cars2 in search path
attach(cars2);
## Create summary data sets for Highway and city gas mileage MPG by number of
cylinders and origin.  ;
## Bootstrap resampling used to calculate 95% confidence intervals of mean;
mean.hw <- summarize(MPG_Highway, llist(Cylinders, Origin), smean.cl.boot);
mean.city <- summarize(MPG_City, llist(Cylinders, Origin), smean.cl.boot);
## rename MPG_Highway and MPG_City to common name:  mean.mpg.;
mean.hw2 <- upData(mean.hw, rename=list(MPG_Highway="mean.mpg"));
mean.city2 <- upData(mean.city, rename=list(MPG_City="mean.mpg"));
##create new variable type to label type of mean.mpg;
mean.hw2$type[mean.hw2$mean.mpg>0]<- "Highway"
mean.city2$type[mean.city2$mean.mpg>0]<- "City"
## Vertical combine of the 2 data frames – new dataframe is named all;
all <- rbind(mean.hw2, mean.city2);
```

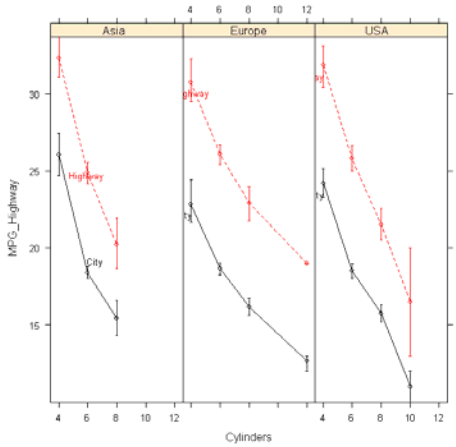
```
##  Use Hmisc function xYplot to plot mean gas mileage and 95% confidence intervals;
## by cylinders with separate panels for each of 3 origins.  ;
## groups – separate curves by type within each panel.;
## col = c(1,2) sets colors for 2 groups (1=black, 2=red). Lty sets line types;
## layout=c(3,1)  determines number of rows and columns – c(3,1) is 3 columns and 1
row;
xYplot(Cbind(mean.mpg, Lower, Upper) ~ Cylinders | Origin, data=all, groups=type,
col=c(1,2), lty=c(1,2), layout=c(3,1), type='o');
endsubmit;
```

A SIMILAR PLOT MADE WITH NEW SGPANEL PROCEDURE

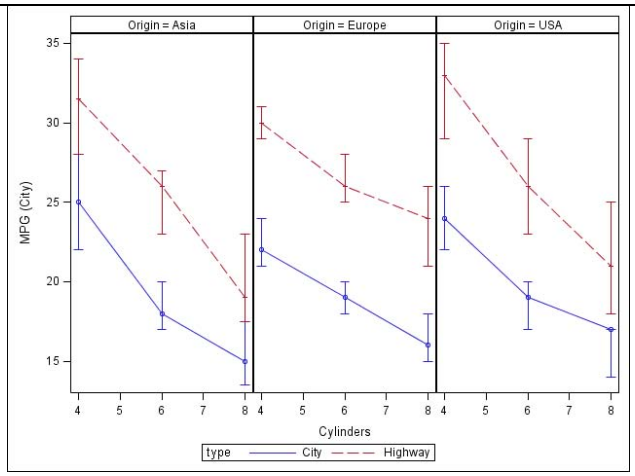
We were pleasantly surprised to learn that we could make a similar graph using the new SGPANEL procedure in SAS. Rather than plot means and confidence intervals again, the SAS graph plots the first, second, and third quartiles (25th percentile, median, and 75th percentile), so the “error bars” are somewhat wider by design. The plot was limited to 4, 6, and 8 cylinder engines, which is slightly different from what was done in R.

```
data cars;
  set sashelp.cars;
  where cylinders in (4, 6, 8);
  run;
proc means data=cars nway;
  var MPG_City ;
  class origin cylinders;
  output out=mn1 median=med q1=q1 q3 =q3;
  run;
  proc means data=cars nway;
  var MPG_Highway;
  class origin cylinders;
  output out=mn2 median=med q1=q1 q3 =q3;
  run;
data means;
  set mn1(in=i) mn2(in=j);
  if i then type="City   ";
  if j then type="Highway";
  run;
proc sgpanel data=means;
panelby origin / rows=1;
series x=cylinders y=med /group=type;
scatter x=cylinders y=med /YERRORLOWER=q1 YERRORUPPER=q3 group=type;
run;
```

FIGURE 2: TRELLIS PLOTS OF CITY AND HIGHWAY GAS MILEAGE BY ORIGIN PRODUCED BY THE XYPLOT() FUNCTION IN R AND THE SGPLANEL PROCEDURE IN SAS.



R graph produced using the xYplot function from HMISC package. Plot of means and 95% confidence intervals for the mean of highway and city gas mileage by origin and number of cylinders of engine.



Graph built with SAS SGPANEL procedure – plot of quartiles (q1, median, q3) of highway and city miles per gallon by origin and number of cylinders of engine.

EXAMPLE 3 – LOGISTIC REGRESSION.

SAS certainly has excellent capabilities to fit logistic regression models. This example was motivated by a problem that we encountered due to quasi-complete separation. The original problem was a logistic regression model with 8 candidate predictors. Paul Allison discussed the problem of quasi-complete and complete separation and how it leads to no solution with standard maximum likelihood estimation, in his 2008 SAS Global Forum paper (PD Allison 360-2008: *Convergence Failures in Logistic Regression*). He recommended Firth logistic regression as a promising solution to the problem. For this example, only the problematic variable is used as a predictor. This example highlights the lack of warnings in some R functions. For example, when fitting a logistic regression model to a data set that had quasi complete separation, SAS LOGISTIC provides warnings that maximum likelihood estimation failed. For the same data set and model, R did not provide any warning that there was a problem. Note that the R output is converted to tables using Microsoft Excel®.

```
> table(death, x1)
```

	death=No	death=yes
x1=no	27	0
x1=yes	24	15

Note, when x1=No, no events occurred. The Logistic Regression Model was fit using the R command:

```
lrm(formula = death ~ x1, data = data1)
```

	Coef	S.E.	Wald Z	P
Intercept	-0.1178	0.2805	-0.42	0.6746
X1	9.3693	26.361	0.36	0.7223

There is no warning message indicating that there was a problem with the estimation! According to the p-value (p=0.7223), we would conclude that X1 has no effect on death.

The same model was run using the LOGISTIC procedure in SAS

```
proc logistic data=data1 ;
  class  x1(ref='0')  /param=ref;
  model  death(event='1') =  x1;
 run;
```

SAS provided several warnings in both the log and listing:

- NOTE: PROC LOGISTIC is modeling the probability that Death=1.
- WARNING: There is possibly a quasi-complete separation of data points. The maximum likelihood estimate may not exist.
- WARNING: The LOGISTIC procedure continues in spite of the above warning. Results shown are based on the last maximum likelihood iteration. Validity of the model fit is questionable.

Once you realize there's a problem, you can use either the exact or Firth penalized likelihood options to refit the model more appropriately. For this simplified version of the problem the exact option is reasonable. With many regressors, the Firth option provides a nice alternative solution as discussed in Allison's paper.

```
**Firth penalized logistic regression – new in SAS 9.2 **;
proc logistic data=data1 ;
  class x1(ref='0') /param=ref;
  model  died (event='1') =  x1  / lackfit FIRTH CLODDS=PL;
   run;

**exact logistic regression in SAS
proc logistic data=data1 EXACTONLY ;
  class  x1(ref='0')   /param=ref;
  model  died(event='1') =  x1 ;
         exact x1 /estimate=both;
            run;
```

IN R, THE FIRTH LOGISTIC REGRESSION CAN BE FIT USING THE LOGISTF PACKAGE.

```
library(logistf, T);
logistf(died ~ x1, data = data1)
Model fitted by Penalized ML
Confidence intervals and p-values by Profile Likelihood
```

	coef	se(coef)	lower 0.95	upper 0.95	Chisq	p
(Intercept)	-0.11551	0.280523	-0.66626	0.429492	0.173173	6.77E-01
X1	3.549499	1.510253	1.451006	8.417985	15.77516	7.13E-05

Likelihood ratio test=15.77516 on 1 df, p=7.133294e-05, n=66

Note that the p-value for x1 is highly significant. The odds ratios for x1 can be calculated by exponentiation the coefficients. The resulting odds ratio based on profile likelihood is 34.8 (95% CI: 4.3, 4528)

```
> print("odds ratios and 95% CI")
[1] "odds ratios and 95% CI"
> exp(g3$coef)
(Intercept)       x1
  0.8909091  34.7958949
> exp(g3$ci.lower)
(Intercept)       x1
  0.5136254   4.2674074
> exp(g3$ci.upper)
(Intercept)       x1
   1.536477 4527.771846
```

Example 4 – Cumulative incidence curves and Gray tests for competing outcomes.

Competing risks occur when experiencing one event precludes a second event from happening. To demonstrate features available to model competing outcomes in the `cmprsk` package in R, we use a Bone marrow transplant data set described in Copelan et al (1991). This data set is an example data set in the Klein JP and Moeschberger textbook, and the data was downloaded from: http://www.mcw.edu/biostatistics/Faculty/Faculty/JohnPKleinPhD/SurvivalAnalysisBook/DataSetsBothEditions.htm. According to Copelan et al: "Bone marrow transplants (BMT) are a standard treatment for acute leukemia. Risk factors for recovery at time of transplant (TX) include: patient and/or donor age and sex, stage of initial disease, time

from diagnosis to TX. TX can be considered a failure when a patient's leukemia returns (relapse) or when he or she dies while in remission (treatment related death)." We use the `cmprsk` package to estimate cumulative incidence of relapse and death, both overall and by sex. We use Gray's tests to compare the equality of the incidences of death and relapse by sex. The `cmprsk` package in R also allows you to fit Fine Gray competing risks regression models to compare incidences adjusting for multiple risk factors. We use a DATA step in SAS to prepare the data for analysis in R. We create an event indicator, and we calculate the time to the first event – death, relapse, or if no event has yet occurred time to last follow-up. Note that t1 is the time to death or last follow-up. T2 is the time to relapse or last follow-up (or death). The DATA included indicators for death and relapse. If `t1=t2` then the first event is death. If `t1>t2`, then relapse is the first event.

SAS code to read data and create event indicators and time to first event (relapse, death, or last follow-up).:

```
data bmt;
  input grp t1 t2 death relapse disfree ta a TC c tp p  z1 z2 z3 z4 z5 z6 z7 z8 z9
z10;
*** Create code for first event - death or relapse ***:
***  if t1>t2 then relapse occurred before death   *;
*** if t1=t2, then no event or death *;
int_event1 = min(t1, t2);
if t1=t2 then event1 = death;
if t1>t2 then event1 = 2*relapse;
if t1>t2 and relapse=0 then do; int_event1=t1; event1=death; end;
** note there is one oddball in which t1>t2, but relapse=0 and death=1.  It makes no
sense;
format event1c $12.;
if event1=0 then event1c ="Disease-free";
if event1=1 then event1c = "Death";
if event1=2 then event1c = "Relapsed";
cards;
…
```

The SAS data set was converted to a comma delimited file using the SAS Export Wizard.

R code:

```
#load libraries;
library(Hmisc, T);   library(cmprsk, T); library(foreign, T);
klein.bmt <- read.csv("c:/sugi2010/sas_r/klein_bmt.csv")
# Use upData utility function from Hmisc library to rename variables;
bmt <- upData(klein.bmt, rename=list(int_event1="int.event1",z3="male" ),
lowernames=TRUE) ;
#male is coded 0/1.  Make sex a factor to have formatted labels.  ;
bmt$sex <- factor(bmt$male, levels=c(0,1), labels=c("Female", "Male"));
attach(bmt)
print(fit.ci2<-cuminc(int.event1, event1c, sex, cencode="Disease-free"))
```

Output from print statement (results converted to tables using Excel®).

GRAY'S TESTS FOR EQUALITY OF CUMULATIVE INCIDENCE CURVES BY SEX OF PATIENT:

Tests:

	stat	pv	df
Death	0.024303	0.876116	1
Relapsed	1.09324	0.295754	1

Estimates and Variances:

CUMULATIVE INCIDENCE ESTIMATES OF DEATH AND RELAPSE BY SEX AT TIME= 500, 1000, 1500, AND 2000 DAYS.

$est

	500	1000	1500	2000
Female Death	0.246711	0.264803	0.264803	0.264803
Male Death	0.25	0.288125	0.317167	0.317167
Female Relapsed	0.337171	0.355263	0.355263	0.355263
Male Relapsed	0.225	0.27625	0.27625	0.27625

VARIANCES OF CUMULATIVE INCIDENCE ESTIMATES OF DEATH AND RELAPSE BY SEX AT TIME= 500, 1000, 1500, AND 2000 DAYS.

$var

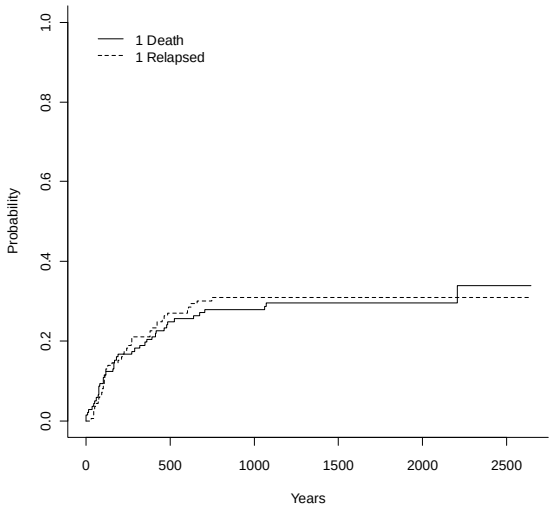
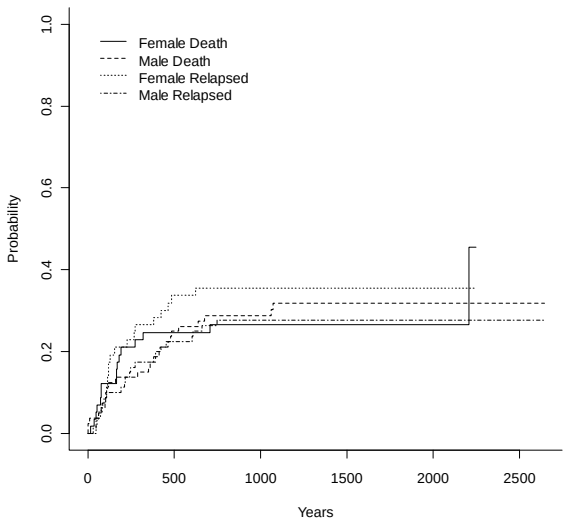
	500	1000	1500	2000
Female Death	0.003363	0.003537	0.003537	0.003537
Male Death	0.002385	0.00262	0.002827	0.002827
Female Relapsed	0.004091	0.004201	0.004201	0.004201
Male Relapsed	0.002217	0.002562	0.002562	0.002562

The `cuminc` function creates estimates of the cumulative incidence of death and relapse by sex of patient. The first table contains results from Gray's tests. Based on Gray's tests, there is no difference in either the cumulative incidence of death (p=0.88) or relapse (p=0.30) by sex of the patient. At 1000 days, the estimated cumulative incidence (i.e. probability) of death is 26.5% for females and 28.8% for males. The cumulative incidence for relapse is 35.5% for females and 27.6% for males (at 1000 days).

R code for cumulative incidence plots:

```
By sex:   plot(fit.ci1<-cuminc(int.event1, event1c, sex, cencode="Disease-free"))
Overall:  plot(fit.ci1<-cuminc(int.event1, event1c,    , cencode="Disease-free"))
```

FIGURE 3A AND 3B: CUMULATIVE INCIDENCE PLOTS FOR DEATH AND RELAPSE – OVERALL AND BY SEX PRODUCED USING THE CMPRSK PACKAGE IN R.

3A: Overall cumulative incidence of death and relapse.	3B: Cumulative incidence of death and relapse by sex
	

CONCLUSION

As statistical consultants, we are faced with the dilemma that no package seems to have every feature that we might want. SAS makes it easier to tap into the features of R with the IML/Workshop. We discussed strategies for using R to complement a SAS analysis. We discussed examples where we have used R to solve problems that we could not have handled easily with SAS alone. SAS is certainly a feature rich system, and we have been impressed by recent improvements and enhancements. Perhaps someday we will be able to submit code to R within a SAS program, to make it even easier to outsource calculations to R.

REFERENCES / RECOMMENDED READING

- Harrell FE: *Regression Modeling Strategies With Applications to Linear Models, Logistic Regression, and Survival Analysis* . New York: Springer, 2001. In third printing.
- Hmisc: Functions for data analysis, high-level graphics, utility operations, computing sample size and power, importing datasets, imputing missing values, advanced table making, variable clustering, character string manipulation, conversion of S objects to LaTeX code, and recoding variables. http://biostat.mc.vanderbilt.edu/wiki/Main/Hmisc
- Design: Functions for regression modeling, testing, estimation, validation, graphics, prediction, and typesetting by storing enhanced model design attributes in the fit. Design is a collection of about 180 functions that assist and streamline modeling, especially for biostatistical and epidemiologic applications. http://biostat.mc.vanderbilt.edu/wiki/Main/Design
- "Calling Functions in the R Language" in *SAS/IML Studio 3.2 for SAS/STAT Users* (http://support.sas.com/rnd/app/studio/statr.pdf).
- Teresa Scott, *Introduction to R notes*. http://biostat.mc.vanderbilt.edu/wiki/pub/Main/TheresaScott/Scott.IntroToR.I.pdf

REFERENCES / RECOMMENDED READING (CONTINUED)

- Cassell, David L. (2007), "Don't Be Loopy: Re-Sampling and Simulation the SAS Way," *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc.
- Fine JP and Gray RJ (1999). *A proportional hazards model for the subdistribution of a competing risk*. JASA 94:496-509.
- Philip R Holland, Holland Numerics Limited, Royston, Herts, UK. SAS to R to SAS. Paper CC03. http://www.hollandnumerics.co.uk/pdf/SAS2R2SAS_paper.pdf
- Paul D. Allison. (2008) "Convergence Failures in Logistic Regression" *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. http://www2.sas.com/proceedings/forum2008/360-2008.pdf
- Klein JP and Moeschberger, M.L. *Survival Analysis: Techniques for Censored and Truncated Data* 2nd Edition, Springer Verlag, New York, 2003.
- Copelan EA , Biggs JC, Thompson JM, Crilley P, Szer J, Klein JP, Kapoor N, Avalos BR, Cunningham I, Atkinson K, Downs K, Harmon GS, Daly MB, Brodsky I, Bulova SI, and Tutschka PJ, *Treatment for acute meyelocytic leukaemia with allogeneic bone marrow transplantation following preparation with BuCy2*, Blood; 78 (3):838-843, 1991.
- Scrucca L, Santucci A, Aversa F. *Competing risk analysis using R: an easy guide for clinicians*. Bone Marrow Transplant 2007; 40: 381–387.
- Carlos Alzola and Frank Harrell. *An Introduction to S and the Hmisc and Design Libraries*. http://biostat.mc.vanderbilt.edu/twiki/pub/Main/RS/sintro.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jeff Gossett

University of Arkansas for Medical Sciences. Department of Pediatrics.

One Children's Way

Little Rock, AR 72202

Work Phone: 501-364-6631

Fax: 501-364-1431

E-mail: GossettJeffreyM@uams.edu

Web: http://www.arpediatrics.org/research/biostatistics